

### AMENDMENTS TO THE CLAIMS

This listing of claims includes a complete listing of both allowed claims and amended claims and will replace all prior versions, and listings, of claims in the application.

#### **Listing of Claims:**

1. (Currently amended): A client side HTTP stack software component embodied in machine readable media and effectuated on a machine that processes requests, comprising:
  - at least one completion port object;
  - a thread pool comprising a plurality of threads that process differentiable tasks associated with at least one client side request; and
  - a client side state machine selectively associated with the at least one request, the client side state machine selected via at least one key associated with the at least one request based at least in part on each differentiable task; and  
an activation component that associates a context of a first thread with the client side state machine using the at least one key and activates the first thread.
2. (Previously presented): The client side HTTP stack implementation of claim 1, further comprising a scheduler thread that activates an object scheduled to begin sending requests at a specific time.
3. (Previously presented): The client side HTTP stack implementation of claim 1, further comprising a DNS thread that resolves domain names into IP addresses.
4. (Previously presented): The client side HTTP stack implementation of claim 1, further comprising a timeout thread with a list of active sockets and timers associated with each socket, the timeout thread selectively times-out at least one socket according to at least one timer in the list.

5. (Previously presented): The client side HTTP stack implementation of claim 4, further comprising a scheduler thread that activates an object scheduled to begin sending requests at a specific time.

6. (Previously presented): The client side HTTP stack implementation of claim 5, further comprising a DNS thread that resolves domain names into IP addresses.

7. (Previously presented): The client side HTTP stack implementation of claim 4, further comprising a DNS thread that resolves domain names into IP addresses.

8. (Currently amended): A machine effectuated software component included on machine readable media that implements a client side HTTP stack, comprising:

a thread pool comprising N threads that process M requests from a client application component, where N and M are integers greater than 1 and where M is greater than N; and

a state machine associated with each of the M requests based at least on one or more tasks included as a part of each of the M requests;

at least one key associated with at least one of the M requests, wherein a first one of the N threads is associated with the at least one of the M requests, and

a thread activation component that associates the context of the first one of the N threads with the state machine using the at least one key, in order to activate the first one of the N threads.

9. (Currently amended): The software component of claim 8, ~~further comprising the~~ at least one thread activation component that activates at least one of the N threads based on an event.

10. (Previously presented): The software component of claim 9, where the at least one thread activation component is a completion port.

11. (Previously presented): The software component of claim 9, where at least one of the N threads deactivates itself and returns to the thread pool when an operation being processed by the at least one of the threads is pending.

12. (Previously presented): The software component of claim 11, where the event is the receipt of a completion packet by the at least one thread activation component.

13. (Previously presented): The software component of claim 12, where the at least one thread activation component is a completion port.

14. (Previously presented): The software component of claim 13, further comprising a scheduler thread that activates an object scheduled to begin sending requests at a specific time.

15. (Previously presented): The software component of claim 14, further comprising a DNS thread that resolves domain names into IP addresses.

16. (Previously presented): The software component of claim 15, further comprising a timeout thread with a list of active sockets and timers associated with each socket, the timeout thread selectively times-out at least one socket according to at least one timer in the list.

17. (Cancelled).

18. (Cancelled).

19. (Currently amended): The software component of claim 8 [[18]], where the thread activation component associates the context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event.

20. (Previously presented): The software component of claim 8, further comprising a scheduler thread that activates an object scheduled to begin sending requests at a specific time.

21. (Previously presented): The software component of claim 8, further comprising a DNS thread that resolves domain names into IP addresses.

22. (Previously presented): The software component of claim 8, further comprising a timeout thread with a list of active sockets and timers associated with each socket, the timeout thread selectively times-out at least one socket according to at least one timer in the list.

23. (Currently amended): A method effectuated at least in part by a machine for implementing a client side HTTP stack, comprising:

processing M requests from a client application component using a thread pool comprising N threads, where M and N are integers greater than 1 and where M is greater than N; [[and]]

selectively associating a state machine with each of the M requests based at least in part on one or more differentiable task included in each of the M requests;

associating at least one key with at least one of the M requests;

associating a first one of the N threads with the at least one of the M requests; and

associating a context of the first one of the N threads with the state machine using the at least one key, in order to deactivate the first one of the N threads.

24. (Original): The method of claim 23, further comprising:

selectively deactivating at least one of the N threads; and

activating at least another of the N threads based on an event using at least one thread activation component.

25. (Previously presented): The method of claim 24, where the at least one thread activation component is a completion port.

26. (Previously presented): The method of claim 24, where selectively deactivating at least one of the N threads comprises deactivating the at least one of the N threads when an operation being processed by the at least one of the N threads is pending.

27. (Previously presented): The method of claim 26, where activating at least another of the N threads based on an event comprises:

receiving a completion packet using the thread activation component; and  
activating one of the N threads upon receipt of the completion packet using the thread activation component.

28. (Previously presented): The method of claim 27, where the at least one thread activation component is a completion port.

29. (Original): The method of claim 28, further comprising activating an object scheduled to begin sending requests at a specific time using a scheduler thread.

30. (Original): The method of claim 29, further comprising resolving domain names into IP addresses using a DNS thread.

31. (Original): The method of claim 30, further comprising selectively timing out at least one socket according to at least one timer associated with the at least one socket using a timeout thread comprising a list of active sockets and timers associated with each socket.

32. (Cancelled).

33. (Cancelled).

34. (Currently amended): The method of claim 23 ~~[[33]]~~, further comprising associating a context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event.

35. (Currently amended): A computer-readable medium having computer-executable instructions for processing M requests from a client application component using a thread pool comprising N threads, where M and N are integers greater than 1 and where M is greater than N, and associating a state machine with at least one of the M requests, the state machine selectively associated based on at least a task included in each of the M requests, the state machine activates at least one of the N threads based at least in part on the task, the computer executable instructions further include:

associating at least one key with the at least one of the M requests;

associating a first one of the N threads with the at least one of the M requests; and

associating a context of the first one of the N threads with the state machine

employing the at least one key, in order to deactivate the first one of the N threads.

36. (Original): The computer-readable medium of claim 35, further comprising computer-executable instructions for:

selectively deactivating at least one of the N threads; and

activating at least another of the N threads based on an event using at least one thread activation component.

37. (Previously presented): The computer-readable medium of claim 36, where the at least one thread activation component is a completion port.

38. (Previously presented): The computer-readable medium of claim 36, where the computer-executable instructions for selectively deactivating at least one of the N threads comprises computer-executable instructions for deactivating the at least one of the N threads when an operation being processed by the at least one of the N threads is pending.

39. (Previously presented): The computer-readable medium of claim 38, where the computer-executable instructions for activating at least another of the N threads based on an event comprises computer-executable instructions for:

receiving a completion packet using the thread activation component; and

activating one of the N threads upon receipt of the completion packet using the thread activation component.

40. (Original): The computer-readable medium of claim 39, further comprising computer-executable instructions for activating an object scheduled to begin sending requests at a specific time using a scheduler thread.

41. (Original): The computer-readable medium of claim 40, further comprising computer-executable instructions for resolving domain names into IP addresses using a DNS thread.

42. (Original): The computer-readable medium of claim 41, further comprising computer-executable instructions for selectively timing out at least one socket according to at least one timer associated with the at least one socket using a timeout thread comprising a list of active sockets and timers associated with each socket.

43. (Cancelled).

44. (Cancelled).

45. (Currently amended): The computer-readable medium of claim 35 ~~[[44]]~~, further comprising computer-executable instructions for associating a context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event.

46. (Currently amended): A machine executed software component resident on machine readable media for implementing a client side HTTP stack, comprising:

means for processing M requests from a client application component using a thread pool comprising N threads, where M and N are integers greater than 1 and where M is greater than N; [[and]]

means for assigning each of the M requests with a state machine, the assignment of the state machine based on one or more differentiable tasks that comprises each of the M requests; and

means for associating at least one key with at least one of the M requests, a first one of the N threads with the at least one of the M requests, and a context of the first one of the N threads with the state machine using the at least one key, in order to deactivate the first one of the N threads.

47. (Original): The software component of claim 46, further comprising:

means for selectively deactivating at least one of the N threads; and

means for activating at least another of the N threads based on an event.

48. (Original): The software component of claim 47, further comprising

means for activating an object scheduled to begin sending requests at a specific time.

49. (Original): The software component of claim 47, further comprising

means for resolving domain names into IP addresses.

50. (Original): The software component of claim 47, further comprising

means for selectively timing out at least one socket according to at least one timer associated with the at least one socket.